Homework #3

CS520 / KAIST Fall 2015

Due: November 11, 2015 2:30PM

1. The fixpoint combinatory **Y** in the class note is define by Curry:

   $\mathbf{Y} \equiv \lambda f.(\lambda x.f(x\ x))(\lambda x.f(x\ x))$.

   There are other fixpoint combinators. One is defined by Turing:

   $\mathbf{Y}_t \equiv (\lambda f.\lambda x.x(f\ f\ x))(\lambda f.\lambda x.x(f\ f\ x))$

   Show that $\mathbf{Y}_t$ is a fixpoint combinator: $\mathbf{Y}_t\ F \to^* F(\mathbf{Y}_t\ F)$

2. Write the compiler (compilation rules) from the applicative language to the (S,E,C,D)-machine language. Semantics of the given source language are already explained in the class (TP Class 6). When you translate pairs (Pair is a syntactic sugar as you know.) of source language, you may compile the source program after you translate pairs into de-sugared form. Or you may invent new machine primitives to implement the pairs. When you use your (S,E,C,D)-machine primitives, describe the semantics using transition rules. Briefly sketch how to prove your compiler implements the semantics of the source language correctly for each language construct.

   Source Languages :

e	$::==\ n$	; integer
	$\mid\ x$	; variable
	$\mid\ e + e$	; primitive integer operation
	$\mid$ **if0** $e\ e\ e$	; branch
	$\mid$ **fn** $x\ e$	; $\lambda x.e$ abstraction
	$\mid$ **rec** $f\ e$	; $\mathtt{rec}\ f\ \lambda x.e'$ recursive function

| **e e**	; application
| **let x e e**	; `let x:=e in e`
| **(e , e)**	; pair
| **fst e**	; `e.1` the first element of pair
| **snd e**	; `e.2` the second element of pair

3. Consider the following machine for evaluating λ-expression by the <u>normal-order evaluation</u> (⇒). Machine consists of three components (e, S, E) where

$e \in \lambda\text{-Expr}$

$S \in \lambda\text{-Expr}^*$; <u>stack</u>

$E \in \mathrm{Id} \to^{\mathrm{fin}} \lambda\text{-Expr}$; <u>environment</u>

The transition $(e, S, E) \to (e', S', E')$ of the machine is define as

$(x, S, E) \to (E(x), S, E)$

$(\lambda x.e, e' \cdot S, E) \to$?

$(e_1\ e_2, S, E) \to$?

Fill out the two question marks, so that for any λ-expression e and a normal term e',

$$e \Rightarrow^* e' \quad \text{iff} \quad (e, \varepsilon, \{\}) \to^* (e', S, E).$$

4. Using the applicative language in TP(class 6), show the proof tree for the following judgment.

{foo → <rec f λx. (if0 x 1 (f 0)), ϕ> } ├ foo 2 => *v*